



# Comparaison et Partitionnement de Séries Temporelles Basés sur la Forme des Séries

Brieuc Conan-Guez, Alain Gély, Lydia Boudjeloud-Assala, Alexandre  
Blansché

## ► To cite this version:

Brieuc Conan-Guez, Alain Gély, Lydia Boudjeloud-Assala, Alexandre Blanché. Comparaison et Partitionnement de Séries Temporelles Basés sur la Forme des Séries. XXVIe Rencontres de la Société Francophone de Classification, Sep 2019, Nancy, France. hal-02971348

**HAL Id: hal-02971348**

**<https://hal.science/hal-02971348>**

Submitted on 19 Oct 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Comparaison et Partitionnement de Séries Temporelles Basés sur la Forme des Séries

Brieuc Conan-Guez, Alain Gély, Lydia Boudjeloud-Assala, Alexandre Blansché

Université de Lorraine, CNRS, LORIA, F-57000 Metz, France  
{brieuc.conan-guez, alain.gely, lydia.boudjeloud-assala, alexandre.blansche}  
@univ-lorraine.fr

**Résumé.** Dans ce travail, nous nous intéressons à la classification non supervisée de séries temporelles. La méthode de partitionnement utilisée, dérivée des centres mobiles, s'appuie sur la forme des séries pour évaluer leurs ressemblances. Afin de comparer ces formes, nous proposons deux mesures de ressemblance entre séries temporelles invariantes par translation et par changement d'échelle. La première mesure est une adaptation de la mesure cosinus pour laquelle l'alignement temporel optimal entre deux séries est obtenu en testant toutes les translations d'une série par rapport à l'autre. La seconde mesure est une version *soft* de la première : le calcul du min sur les différents alignements est effectué grâce à la fonction *softmin*.

## 1 Introduction

La classification de séries temporelles est une problématique majeure depuis de très nombreuses années. Dans ce travail, nous nous intéressons spécifiquement à des méthodes de partitionnement qui effectuent une comparaison des séries temporelles basée sur un critère de forme. Les mesures de ressemblance proposées sont doublement invariantes : elles ne dépendent pas de l'amplitude des séries (normalisation des valeurs des séries), et sont invariantes par décalage temporel. Contrairement à la mesure de déformation temporelle dynamique (DTW), qui effectue une déformation non linéaire du temps (le meilleur alignement est obtenu par contraction ou dilatation locale de l'axe temporel), les mesures de ressemblance auxquelles on s'intéresse dans ce travail effectuent une translation uniforme de l'axe temporel. Ce type de mesures s'applique naturellement dans le cas où l'origine temporelle des séries est inconnue, mal définie ou différente entre séries : signaux périodiques, signaux pour lesquels l'outil de mesure peut induire un décalage sur l'axe des  $x$  (données spectrométriques), signaux pour lesquels la captation des données débute après l'origine du phénomène. Ce dernier cas se retrouve par exemple lors de l'analyse du cycle de vie de hashtags apparaissant sur des réseaux socio-numériques : la période de captation des tweets citant le hashtag débute après l'apparition du premier tweet mentionnant le hashtag.

Plusieurs méthodes de partitionnement s'appuyant sur ce type de mesures ont déjà été proposées par le passé. On peut citer K-Spectral Centroid (Yang et Leskovec (2011)) et K-Shape (Paparrizos et Gravano (2017)). Ces méthodes ont été éprouvées sur de nombreux jeux de données (Paparrizos et Gravano (2017)) et comparées avec de nombreuses méthodes comme

K-MeansDTW (DTW Barycenter Averaging). Les méthodes K-Spectral Centroid (KSC) et K-Shape (KS) ont un coût de calcul assez important, évoluant cubiquement avec la longueur des séries  $L$ . Certaines astuces de calcul (Conan-Guez et al. (2018)) permettent malgré tout d'observer dans la pratique un comportement quadratique.

Dans ce travail, nous proposons deux nouvelles mesures de ressemblance entre séries temporelles. La première est similaire à celles de KSC et KS, et s'appuie sur la mesure cosinus qui assure l'invariance par changement d'échelle sur l'axe Y. L'invariance par décalage temporel est obtenue en testant tous les décalages de la seconde série par rapport à la première afin d'identifier l'alignement temporel optimal. La méthode de partitionnement basée sur cette première mesure se révèle très rapide avec une complexité sous-quadratique  $O(L \log(L))$  en la longueur des séries  $L$ . La seconde mesure est la version régularisée de la première : le calcul du minimum apparaissant dans la recherche de l'alignement temporel optimal est effectué cette fois grâce à la fonction softMin. Bien que cette seconde mesure soit plus coûteuse à évaluer, elle présente l'avantage d'être régulière, et par la même peut être adaptée à des méthodes de fouille de données nécessitant le calcul de gradients (méthodes neuronales par exemple).

Les contributions de cet article sont donc les suivantes : nous proposons deux mesures de ressemblance basées sur la forme pour la comparaison de séries temporelles. Nous montrons que ces mesures mais surtout le gradient de la seconde mesure peuvent être évalués en temps sous-quadratique ( $O(L \log(L))$ ). Nous proposons une analyse de KSC, KS et des méthodes de partitionnement proposées dans ce travail.

## 2 Mesures de ressemblance et méthodes de partitionnement

### 2.1 Mesures de ressemblance basées sur la forme

On considère ici des séries temporelles de signes quelconques. Les séries ont toutes la même longueur, notée  $L$ . On considère l'opérateur de décalage temporel  $\tau_o$  de paramètre  $o$  ( $-L + 1 \leq o \leq L - 1$ ). La corrélation croisée normalisée prend la forme suivante :

$$CCN(x, y)(o) = \frac{x \cdot \tau_o(y)}{\|x\| \|y\|}$$

la notation "point" correspond au produit scalaire, et  $\|x\|^2 = x \cdot x$  (norme L2).

On note  $g_E$  la fonction  $g_E(cc) = \sqrt{1 - cc}$ . On rappelle que la distance euclidienne entre  $x$  et  $y$ , des séries normalisées, est égale à  $g_E(CCN(x, y)(0))$  à un coefficient multiplicatif près.

La première mesure  $d_E$  que nous proposons prend la forme suivante :

$$d_E(x, y) = \min_o g_E(CCN(x, y)(o))$$

Les mesures utilisées par KSC et KS sont très similaires. Seule la nature de la fonction  $g$  change : pour KS, on a  $g_{KS}(cc) = 1 - cc$ , alors que pour KSC, on a  $g_{KSC}(cc) = \sqrt{1 - cc^2}$ . Notons qu'à cause de l'élévation au carré intervenant sur  $cc$ , les séries corrélées et anti-corrélées sont considérées comme identiques par  $d_{KSC}$ . C'est la raison pour laquelle la méthode KSC a initialement été proposée pour des séries positives (distributions temporelles).

Le calcul de  $CCN$  pour toutes les valeurs de décalage  $o$  semble a priori quadratique en  $L$ . Mais classiquement, on peut ramener cette complexité à  $O(L \log(L))$  en évaluant tous les

produits scalaires  $x \cdot \tau_o(y)$  grâce à la transformée de Fourier rapide FFT. Si l'on note  $\mathcal{F}$  (resp.  $\mathcal{F}^{-1}$ ) la transformée de Fourier (resp. l'inverse de la transformée de Fourier), et  $z^*$  le conjugué du complexe  $z$ , la corrélation croisée de  $x$  et  $y$  est proportionnelle à  $\mathcal{F}^{-1}(\mathcal{F}(x^0) \mathcal{F}^*({}^0y))$ .  $x^0$  (resp.  ${}^0y$ ) est le vecteur  $x$  complété à droite (resp. à gauche) par des zéros. Les auteurs de KS utilisent cette astuce mathématique pour accélérer l'évaluation de leur mesure. KSC et notre mesure  $d_E$  peuvent de même bénéficier de cette remarque.

La seconde mesure de ressemblance  $d_E^\gamma$  que nous proposons s'inspire des travaux portant sur le SoftDTW (Cuturi et Blondel (2017)). Ces travaux utilisent la fonction softMin pour évaluer la ressemblance entre deux séries après déformation temporelle. Les auteurs montrent que le calcul du gradient de SoftDTW peut être réalisé efficacement.

Soit  $g$  un vecteur de longueur  $2L - 1$ , on note  $\min^\gamma(g)$  la fonction SoftMin corrigée :

$$\min^\gamma(g_1, \dots, g_{2L-1}) = -\gamma \log \sum_o e^{-\frac{g_o}{\gamma}} + \gamma \log(2L - 1) \quad \text{pour } \gamma > 0$$

La fonction  $\min^\gamma$  corrigée est une f-moyenne généralisée. Elle vérifie la propriété  $\min(g) \leq \min^\gamma(g) \leq \min(g) + \gamma \log(2L - 1)$ . On voit donc que  $\min^\gamma$  approche la fonction min quand le paramètre de régularité  $\gamma$  tend vers 0.  $\min^\gamma$  est d'autant plus régulière que  $\gamma$  est grand. Elle est dérivable pour  $\gamma$  strictement positif.

Grâce à la fonction SoftMin, il est aisé de définir la version régularisée de  $d_E$ , notée  $d_E^\gamma$  :

$$d_E^\gamma(x, y) = \min_o g_E(CCN(x, y)(o))$$

$d_E^\gamma$  est à valeurs positives mais ne vérifie par la propriété de séparation ( $d_E^\gamma(x, x) \neq 0$ ). L'évaluation de  $d_E^\gamma$  peut s'appuyer elle aussi sur la FFT pour une complexité sous-quadratique ( $O(L \log(L))$ ). Montrons à présent que le gradient en  $x$  de  $d_E^\gamma(x, y)$  peut aussi se calculer efficacement. Après quelques calculs, on obtient le résultat suivant :

$$\frac{\partial d_E^\gamma(x, y)}{\partial x_i} = \frac{\sum_o S_o^\gamma g'_E(CCN(x, y)(o)) y_{i-o}}{\|x\| \|y\|} - \frac{x_i}{\|x\|^2} \sum_o S_o^\gamma g'_E(CCN(x, y)(o)) CCN(x, y)(o)$$

avec  $g'_E$  la dérivée de  $g_E$ ,  $S_o^\gamma = SM_o(-g_E(CCN(x, y)/\gamma))$  où  $SM_o(z) = \frac{e^{z_o}}{\sum_{o'} e^{z_{o'}}$  est la fonction exponentielle normalisée.

Les  $CCN(x, y)(o)$  se pré-calculent en  $O(L \log(L))$ . Le premier terme de la dérivée partielle fait intervenir un produit de convolution, et se calcule donc pour tous les  $i$  en  $O(L \log(L))$  grâce à la FFT. Le second terme est en temps linéaire, car la somme sur  $o$  peut être précalculée (indépendante de  $x_i$ ). L'évaluation de la mesure  $d_E^\gamma$  et de son gradient ont donc des complexités identiques  $O(L \log(L))$ . Afin d'éviter les débordements numériques dans les calculs, on utilise pour les deux évaluations l'astuce classique de décalage des valeurs (*Log-Sum-Exp trick*).

## 2.2 Extraction d'un barycentre

On note KE (resp. KSE) la méthode des centres mobiles basée sur  $d_E$  (resp.  $d_E^\gamma$ ). Comme l'extraction d'un barycentre est utilisée ici comme procédure interne aux centres mobiles, KSC, KS et KE font l'hypothèse que l'alignement optimal  $o_i^*$  de chaque série  $x_i$  avec le barycentre produit à l'itération précédente a été trouvé lors de la phase d'affectation. Les trois méthodes effectuent donc l'extraction du barycentre avec un décalage temporel fixé.

Bien que KSC et KS ne partagent pas la même mesure de ressemblance, l'extraction du barycentre  $\mu$  s'effectue en résolvant le même problème d'optimisation. KSC optimise un critère d'inertie cohérent avec la phase d'affectation :  $\mu = \arg \min_{\mu} \sum_i g_{KSC}^2(CCN(\mu, x_i)(o_i^*))$ . KS optimise le critère équivalent  $\mu = \arg \max_{\mu} \sum_i CCN^2(\mu, x_i)(o_i^*)$ , qui diffère de celui optimisé lors de sa phase d'affectation. Dans le cas de séries à signes quelconques, on remarque que l'extraction du barycentre est susceptible de ne pas distinguer les séries corrélées des séries anti-corrélées ( $CCN$  au carré). Dans la pratique, cette remarque a peu de conséquences, car lors de l'affectation,  $d_{KS}$  choisit les  $o_i^*$  et les classes qui maximisent  $CCN$ .

Pour KSC comme pour KS, le calcul du barycentre  $\mu$  revient à extraire le vecteur propre associé à la plus grande valeur propre d'une matrice (occupation mémoire en  $O(L^2)$ ). Le signe du barycentre doit être déterminé car un vecteur propre peut avoir deux directions. KSC et KS effectuent une diagonalisation complète de la matrice (complexité cubique). Nous avons montré dans un précédent travail (Conan-Guez et al. (2018)), qu'en utilisant un calcul incrémental de la matrice et la méthode des puissances itérées, la complexité observée de KS et KSC était quadratique en  $L$ , d'où une réduction importante des temps de calcul.

Pour KE, la phase d'affectation fournit exactement la même partition que celle de KS, en revanche, l'extraction du barycentre est plus simple. En effet, la mesure  $d_E$  une fois  $o_i^*$  fixé correspond à la distance euclidienne entre les vecteurs unitaires  $\mu$  et  $\tau_{o_i^*}(x_i)$ . On résout  $\mu = \arg \max_{\mu} \frac{\mu}{\|\mu\|} \cdot (\sum_i \frac{\tau_{o_i^*}(x_i)}{\|x_i\|})$ .  $\mu$  est donc le vecteur (rendu unitaire) obtenu en sommant les vecteurs  $\frac{\tau_{o_i^*}(x_i)}{\|x_i\|}$ . L'extraction du barycentre est donc très semblable à celle des centres mobiles classiques avec une complexité linéaire en  $L$ . KE reste pour autant en  $O(L \log(L))$  du fait de la phase d'affectation. L'occupation mémoire est en  $O(L)$ , car la FFT est en  $O(L)$ .

Pour KSE, le barycentre s'obtient par  $\mu = \arg \min_{\mu} \sum_i d_E^\gamma(\mu, x_i)^2$ . Le carré est ici maintenu afin de résoudre un problème identique à celui de KE. KSE est la seule méthode qui prend en compte l'alignement temporel dans la phase d'affectation comme dans la phase de représentation. Le fait que  $d_E^\gamma$  soit toujours à valeurs positives permet d'avoir un critère d'inertie bien formé. Extraire le barycentre nécessite cette fois l'utilisation d'une méthode d'optimisation non linéaire (gradient conjugué, BFGS,...). Cette méthode ne nous assure pas l'obtention d'un optimum global (contrairement aux méthodes précédentes). En revanche, d'une itération à l'autre du partitionnement, le déplacement continu des barycentres doit favoriser un nombre réduit de descentes de gradient. Pour la méthode du gradient conjugué, si l'on suppose une indépendance entre la dimension  $L$  et le nombre d'évaluations du critère d'inertie, l'occupation mémoire est linéaire en  $L$  et le coût de calcul est en  $O(L \log(L))$  grâce à l'évaluation efficace de  $d_E^\gamma$  et de son gradient.

### 3 Expériences

Les différentes méthodes ont été implémentées avec le langage R. Le critère d'arrêt porte sur l'évolution du critère d'inertie intra-classe. Nous utilisons les fonctions `optim("BFGS")` et `powerMethod` (package `matlib`) pour l'extraction des barycentres de KSE et KS. KS et KE ont été testés avec ou sans centrage des barycentres (Paparrizos et Gravano (2017)).

Afin d'illustrer l'invariance temporelle des méthodes, la figure 1 montre le résultat du partitionnement produit par KE sur un jeu de données artificielles. Ce jeu de données est constitué de 9 séries : 3 gaussiennes, 3 fonctions créneaux et 3 fonctions sinus. Les séries sont déphasées

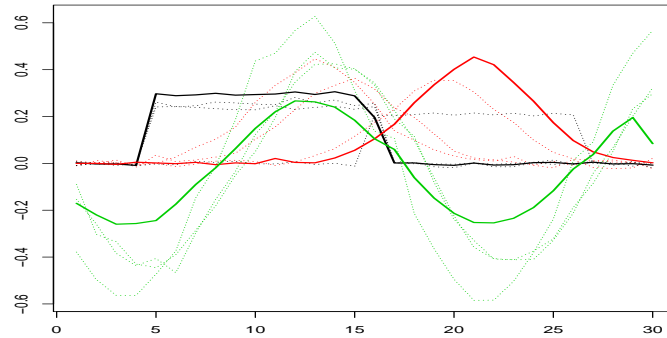


FIG. 1 – Les séries (pointillés) et les barycentres (lignes) colorés selon la classe

(décalage temporel) avec un changement d'échelle (amplitude) et bruitées. Les trois classes sont parfaitement retrouvées par KE.

Nous avons comparé KS, KE et KSE sur 5 jeux de données issus de l' *UCR Time Series Classification Archive*. La table 1 contient une description de chaque jeu de données. Chaque jeu est fourni avec un partitionnement réel (colonne "nb classes") qui sera comparé avec les partitionnements produits par les trois méthodes KS, KE et KSE.

	nb séries	longueur	nb classes
ECG5000	500	140	5
WordsSynonyms	267	270	25
Trace	100	275	4
Fish	175	463	7
Haptics	155	1 092	5

TAB. 1 – Description des cinq jeux de données

Comme nous ne pouvons pas comparer les critères d'inertie intra-classe (ils sont calculés avec des mesures différentes), nous utilisons, comme les auteurs de KS, le Rand-Index pour mesurer l'adéquation entre partition réelle et partition produite. Chaque méthode a été exécutée 10 fois. Les initialisations sont identiques pour toutes les méthodes. La table 2 indique la moyenne des 10 Rand-Index et le temps cumulé des 10 exécutions en secondes.

	ECG5000	WordsSynonyms	Trace	Fish	Haptics
KS	0.750 (59)	<b>0.892</b> (176)	0.785 (12)	0.788 (55)	<b>0.676</b> (220)
KE	<b>0.761</b> (67)	0.891 (185)	0.785 (11)	<b>0.789</b> (38)	0.674 (43)
KSE $\gamma = 0.01$	0.744 (107)	0.879 (291)	<b>0.791</b> (21)	0.777 (17)	0.666 (249)

TAB. 2 – moyenne des 10 Rand-Index (temps cumulé en secondes)

On constate que les performances de KS et KE sont très semblables. KSE est légèrement en retrait, ceci s'explique sans doute par l'existence d'optima locaux. Les résultats pour  $\gamma$  valant 0.1 et 0.001 sont relativement identiques à ceux présentés pour  $\gamma = 0.01$ . Pour des valeurs plus grandes ( $\gamma = 1$ ), les résultats se dégradent (Rand-Index moyen de 0.384 pour Fish par exemple). Si l'on calcule la moyenne des Rand-Index sur les cinq jeux de données, on obtient 0.769 pour les centres mobiles classiques, 0.772 pour KSE, 0.775 pour K-MeansDTW (sans et avec fenêtre 10%), 0.778 pour KS et 0.780 pour KE.

Une analyse des temps de calcul peut aussi être menée, même si le code R n'a pas été écrit dans un souci de performance optimale. Pour des longueurs faibles ( $L < 300$ ), KE et KS ont des temps d'exécution identiques : bien que KS ait une complexité observée quadratique alors que KE soit en  $O(L \log(L))$ , les temps de calcul sont dominés par la phase d'affectation qui est identique pour les deux méthodes. Pour Haptics ( $L = 1092$ ), KE se révèle cette fois 5 fois plus rapide que KS. KSE est bien sûr plus lent dans tous les cas, mais étonnamment la différence n'est pas aussi marquée que l'on aurait pu s'y attendre. Elle s'explique d'une part par la qualité inférieure des solutions, mais aussi peut être par le fait que la phase de représentation de KSE "optimise" l'alignement temporel alors qu'il est maintenu fixe pour KS et KE. KS et KE auraient donc besoin de plus d'itérations (affectation, représentation) pour stabiliser les alignements. Pour ECG5000, KS et KE nécessitent respectivement 138 et 171 itérations (cumulées sur les 10 exécutions), alors que KSE converge en 95 itérations.

Pour conclure, KE est plus simple à implémenter que KS et produit des résultats similaires en un temps plus court. La mesure  $d_E^\gamma$  obtient des résultats honorables et peut être adaptée avantageusement à des méthodes nécessitant des calculs de gradient (méthodes neuronales,...).

## Références

- Conan-Guez, B., A. Gély, L. Boudjeloud-Assala, et A. Blansché (2018). K-spectral centroid : extension and optimizations. In *26th European Symposium on Artificial Neural Networks, ESANN 2018, Bruges, Belgium, April 25-27, 2018*.
- Cuturi, M. et M. Blondel (2017). Soft-DTW : a differentiable loss function for time-series. In *In 34th International Conference on Machine Learning*, Volume 70, pp. 894–903.
- Paparrizos, J. et L. Gravano (2017). Fast and accurate time-series clustering. *ACM Trans. Database Syst.* 42(2), 8 :1–8 :49.
- Yang, J. et J. Leskovec (2011). Patterns of temporal variation in online media. In *Proc. of the fourth ACM international conf. on Web search and data mining*, pp. 177. ACM.

## Summary

In this work, we propose two k-means-like methods, devised for time series clustering. Each method relies on a custom dissimilarity measure between time series, which is invariant to time shifting and Y-scaling. The first measure is an adaptation of the cosine dissimilarity for which the best time alignment is obtained by testing all temporal translations. The second measure is a soft version of the first measure: the min computation on the different time alignments is carried out by the soft min function.